

05-25-00

A

PATENT
NH-1JC759 U.S. PTO
09/578192

05/24/00

In the United States Patent and Trademark Office

Honorable Commissioner of Patents and Trademarks
Washington, DC 20231

Sir:

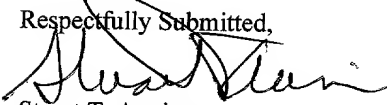
Please file the enclosed patent application papers:

Applicant: Condamoor et al.

Title: Attribute Tree for Modeling Value of Multi-Attribute Products/Services for an Online Trading System

☒ Specification, claims, and abstract: Number of sheets 31☒ Declaration/Oath: Date Signed 5-24-00☒ Drawings: Number of sheets (in triplicate) 10☒ Small Entity Declaration of Assignee.☒ Check for small-entity filing fee of \$ 549 , 37 C.F.R. § 1.16(a). Total # claims 34 , # Independent Claims 5 .☒ Return receipt postcard addressed to applicant's Agent.☒ The commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to deposit account No. **01-2950**.☒ Any additional filing fees required under 37 C.F.R. § 1.16.☒ Any patent application processing fees under 37 C.F.R. § 1.17.☐ The commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to deposit account No. **01-2950**.☐ Any additional filing fees required under 37 C.F.R. § 1.16.☐ Any patent application processing fees under 37 C.F.R. § 1.17.

Respectfully Submitted,



Stuart T. Auvinen

Reg. No. 36,435

Agent for Applicant

Address all inquires to:

Stuart T. Auvinen
429 Twenty-Sixth Ave.
Santa Cruz, CA 95062
(831) 476-5506

Express Mail Label #

EJ133686690US

Date of Deposit

5-24-00JC828 U.S. PTO
05/24/00

05/24/00

PATENT APPLICATION

**Attribute Tree for Modeling Value of Multi-Attribute
Products/Services for an Online Trading System**

INVENTORS:

Ravi V. Condamoor
1382 FisherHawk Drive
Sunnyvale, CA 94087
A citizen of USA

Ankur Datta Sharma
1065 Marigold Court
Sunnyvale, CA 94086
A citizen of USA

Neelakantan Sundaresan
492 Capitol Village Circle
San Jose, CA 95136
A citizen of India

ASSIGNEE:

NehaNet Corp.
2355 Paragon Drive, Suite E
San Jose, CA 95131

Stuart T. Auvinen
429 Twenty-Sixth Ave.
Santa Cruz, CA 95062-5319
(831) 476-5506
Reg. No. 36,435
PTO Customer # 23933



23933

PATENT TRADEMARK OFFICE

PATENT

NH-1

05/23/00 3:14 PM

5 ATTRIBUTE TREE FOR MODELING VALUE OF MULTI-ATTRIBUTE
PRODUCTS/SERVICES FOR AN ONLINE TRADING SYSTEM

RELATED APPLICATION

10 A provisional patent application entitled "System and Method of Modeling Value in an Electronic Trading System" was filed on 4/17/00 by the same inventors for the present application, U.S. Provisional Appl. No. 60/198,125.

FIELD OF THE INVENTION

15 This invention relates to electronic commerce systems, and more particularly to value and pricing of complex products.

BACKGROUND OF THE INVENTION

20 The widespread adoption of the Internet has facilitated new relationships among buyers and sellers. Internet auctions and name-your-price services have empowered the consumer, offering deeper discounts with minimal consumer effort. More recently, business buyers have taken note of the price discounting on the Internet and have been seeking to cut costs through Internet-based purchasing. Business-to-Business (B2B) exchanges have sprung up in many markets to encourage competition among suppliers. Transaction costs are dramatically lowered since paper-based quotes and purchase orders are replaced by electronic documents.

While such electronic commerce exchanges are useful, often the products sold are complex. Many variations or configurations of a product can be sold. In the past, fixed prices were usually assigned to each configuration of a product or each product option, and a buyer merely had to add up the cost of the various options desired. However, these electronic exchanges often do not sell products based on a fixed price schedule. Complex pricing models may be based on volume purchased by a single buyer, or aggregation of demand by many buyers. The price may even be determined by auction, or by some other negotiation between the buyer and seller. The negotiations may even be carried out by computer programs that follow pre-programmed rules.

When complex pricing is combined with many possible configurations of a product, the complexity of any pricing models increases dramatically. When many buyers aggregate demand, each of the buyers may prefer a different configuration of the base product, resulting in a confusing demand mix. Modeling such mixed demand is problematic.

The various product configurations can be described using multiple product attributes. Multi-attribute models are known in relational database and Online Analytical Processing (OLAP) systems. Multi-attribute and multi-attribute electronic commerce systems include multi-attribute auctions, negotiations, and contracts in electronic procurement and other areas.

In "Multi-attribute auctions for electronic procurement", by Bichler, Kaukal, & Segev (IAC's Workshop on Internet Based Negotiation Technologies, March 18-19, 1999 Yorktown Heights, NY), the authors describe a system wherein price is one of the many attributes in determining the value of a product. Other attributes may optionally include shipping-handling costs, warranty etc. Here the value of a product is computed in terms of a utility function which is the weighted sum of the utility units corresponding to each attribute.

While such multi-attribute models are useful, a multi-attribute model more suited for electronic exchanges is desired. A way to express the many options or configurations of products in a product family is desired. It is desired to receive, store, and compare multiple product configurations from many buyers and sellers. A simple yet powerful structure to store many attributes of a product family is desired.

SUMMARY OF THE INVENTION

The present invention describes a system, method, and apparatus for participants in an electronic commerce application to model Value as a Complex Property. Value is modeled using the notion of Attributes.

The system includes mechanisms for

1. Naming Attributes
2. Defining Attribute metadata
3. Giving Attributes Priorities
4. Specifying and working with Dependencies Between Attributes
5. Specifying and working with attribute ranges

Further, the system includes facilities for

1. Representing Attributes
2. Algorithms for Creating, Updating, and Looking up Attributes

Attributes are represented using a dependency tree structure. Each attribute has a set of possible values. Each node in the tree represents an attribute and one of its possible values. The parent of such a node is a node that represents an attribute that this one is dependent on. For an attribute A with possible values vA_1, vA_2, \dots, vA_k and an attribute B that is dependent on it with possible values vB_1, vB_2, \dots, vB_l , each (A, vA_i) node has l child (B, vB_j) $j = 1 \dots l$.

The attribute tree for any trading partner can grow to be exponential. For instance, for a trading partner with N attributes, A_1, A_2, \dots, A_N , and M values for each of the attributes, where some of these attributes are dependent on others, in the worst case is a trading partner tree with $(M^*N - 1)/(M-1)$. A system is provided that can, using

5 heuristics, reduce this size to a manageable size.

In general, a system is provided that allows a trading partner to breakdown and model a commodity in terms of a number of attributes and attribute values. In addition, the modeling system provides a facility to build an augmented products by adding new

10 attributes or deleting existing attributes dynamically during the trade representing market conditions.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a trading partner (TP) attribute Tree.

Figure 2 shows a path from the root to a leaf of an attribute tree.

Figure 3 shows how the data from multiple data sources or applications can be obtained implicitly.

15

Figure 4 is a block diagram of an attribute-tree builder.

Figure 5 shows one embodiment of a heuristic to create an attribute tree.

Figure 6 is a procedure for optimizing an attribute tree.

Figure 7 is a flowchart for a Business-to-Consumer (B2C) application of the attribute trees.

20

Figure 8 is a flowchart of an auction system enhanced by the attribute-structuring facility.

Figure 9 is a flowchart of an optimal product/seller selector that helps a bidder.

Figure 10 illustrates dynamic augmentation of a product with the attributes most desired by the potential buyers.

25

TERMINOLOGY

1. Electronic Commerce – Commerce conducted over an electronic medium
2. Electronic Exchange – A platform in which providers of goods or services buy, sell, and exchange those entities.
- 5 3. Utility Function – defines a function used to define the value of a product. Typically this is price in some units like dollars or pounds that the buyer is willing to buy at or the seller is willing to sell at.
4. Delta or Delta Value - A quantifiable vested interest of a trading partner. Different trading partners usually have different aspects of a trade that they value the most. Example: One might have price, the other may be concerned about quality, and the third might care about quantity and delivery time. The Delta is a manifestation of their preferences and can be positive or negative; positive indicates that the trading partner cares more about it and negative connotes the trading partner is less interested in that particular attribute of a trade.
- 10 5. Product (Product and Service used interchangeably)
6. Base Product Value or Base Value - is the utility function value (typically price) at which a trading partner starts valuing a product.
7. Delta Product Value - is the utility function value (typically price) of the product with the deltas for the attributes accounted for
- 20 8. Trade – A trade is an activity in which one or more parties provide services or goods to one or more parties at a pre-calculated or at a dynamically decided or negotiated cost.
9. Service category – Types of goods and services
10. Trading Partner – A buyer / seller / facilitator in a multi-party trade. Also a service provider.
- 25 11. Net Economic Value – Is the difference between the utility function value (typically price) at which a trading partner values a product and the utility function value (price) he actually trades at.

12. Attribute - An abstract data type that provides a mechanism for Trading Partners to describe their product/service to a finer level of granularity taking into account the strengths and weaknesses of the trading partners. Domain specific attributes will be defined by subclassing the base Attribute class. Ex: Date, Location, Weight are all attributes in the system
13. Attribute Instance - An instance of a valid Attribute. Ex: Instance of a Date attribute.
14. Attribute Value - Quantifiable values assigned to Attribute instances. Ex:)6/01/00 assigned to a Date attribute
15. Independent Attribute – Attribute with values whose delta is independent of any other attribute value
16. Dependent Attribute – Attribute with values whose delta is dependent on another attribute value
17. Attribute Range – Range over which an attribute's values are defined
18. Sets of Ranges – is a collection of disjoint ranges
19. Attribute Interval – defines a segment between two contiguous attribute values in a range
20. Attribute Metadata – Properties of the attribute class (ranges, casting function, creator, etc.) constitute the attribute metadata.
21. Decision Master – The process or object that helps a trading partner decide on participation in one or more trades optionally based upon past trade performances and other business intelligence information.
22. Attribute Value Tree – a dependency tree of attributes and their values for a product where any path from the root to a leaf represents a combination of attribute values chosen for the product.
23. Attribute Value Delta Tree – an Attribute Value Tree which in addition to the attributes and values keeps DELTA values. Delta values may be kept at each node or cumulatively for the paths at each leaf node.
24. TP Tree - A data structure containing the set of <attribute instance, attribute value> pairs and their respective Deltas of a Trading Partner. Though similar in

structure to a Trade Tree, a Trading Partner tree is typically a much larger set as it represents all services and their attributes that affect the final value of the service.

25. Trade Tree - A data structure containing the set of <attribute instance, attribute value> pairs and their respective Deltas for a Trade.

5 DETAILED DESCRIPTION

The present invention relates to an improvement in e-commerce. The following description is presented to enable one of ordinary skill in the art to make and use the invention as provided in the context of a particular application and its requirements. Various modifications to the preferred embodiment will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed.

The Decision Master (DM) is primarily responsible for making a decision on behalf of a Trading Partner (TP) in an Electronic Commerce System. The TP could be a participant in online shopping, auction, and electronic exchanges. The DM works with Attributes that model the notion of Value for the TP.

An Attribute describes the value that a TP gives to a particular aspect of a product. For instance, a customer buying a travel ticket online may measure the value of that product as several attributes like time-of-departure, day-of-departure, month-of-departure, time-of-arrival, day-of-arrival, month-of-arrival, number-of-stops, airline, type-of-aircraft, type-of-seat etc.

Each attribute has values defined over one or more ranges. For instance, number-of-stops may have a range of 0 to 3 with intervals of length 1. Thus, the range 0 to 3 has 3 intervals 0-1, 1-2, 2-3. Attribute values are defined at the highest discrete point in a range which in this case is 1, 2, or 3. It is possible for attributes to have more than one

range. For instance, an airline that flies Monday-Wednesday, Friday-Saturday has 2 ranges with 3 values in the first range and 2 in the second range. It can be assumed that in all intervals in any range are of fixed length. A non-fixed interval length range can be broken down into a collection of intervals with fixed length ranges.

5

Attributes may be dependent or independent. For instance, for each Day-Of-Week attribute value there are several Time-Of-Day attribute values. Here the Time-Of-Day attribute is dependent on the Day-Of-Week attribute. An attribute that does not depend on any other attribute is called an independent attribute. In this discussion one dependence tree structure is created by creating clever or arbitrary dependencies between attributes.

10

The Attribute data structure and the associated operators are shown in the attached code of the provisional application.

15

The system also provides a mechanism to achieve differential pricing based upon attributes and their values. For each attribute value, a Delta is assigned which basically describes what the TP would pay if the product that the TP buys or sells includes this attribute value. The Delta is simply described as a number that could translate to money or any other quantifiable entity. For instance, a traveler may be willing to pay 50\$ more for an airline ticket that involves 0 stops or 50\$ less for a ticket for each stop that the flight involves.

20

Attributes may be Implicit or Explicit. Explicit Attributes are those that are explicitly described by the TP. Implicit Attributes are those that are implicitly discovered by the system based upon data obtained from information gathering or other business intelligence applications.

25

It is assumed that Attributes may be managed, stored, and served by an Attribute Manager (AM) process or Server. For each attribute the AM contains information

30

about the name, ranges, and intervals of the attribute. Note that this AM is not dependent on any particular eCommerce or trading system that the TP is participating in.

The inventors' system creates a TP Attribute tree (TP Tree) which encapsulates the dependency between all the attributes that the TP cares about for a particular product. It is described how the TP Tree is created and managed for a product. The Tree accommodates both dependent and independent attributes for implicit and explicit types of attributes.

A collection of TP trees wherein within each tree the nodes are organized based upon dependency may be merged into a single TP tree. This may be required to enhance searching.

Figure 1 shows a trading partner (TP) attribute Tree. The root 1 of the tree specifies the product that the TP is interested in trading. The base product 16 is an airline ticket from San Francisco to New York. The base value 14 is \$300. Every node 2-10 of the tree represents an attribute name and value. Any path from the root of the tree to a leaf 20 specifies a single product. This path is a path of attribute names and values. In the case of a tree with explicit attributes (where the client has explicitly specified the attributes and values) one could load the root of the tree with the "basic product value"(BPV) that the TP specifies for the product. The edges are labeled with delta-values 12 (+ or -) by which the TP modifies the product value when the subpath from the root 1 to the node at the end of the edge is chosen. For example, in figure 1, the Delta Product Value (DPV) from node 1-2 is +30, while the DPV from node 1-6 is the sum of DPVs from 1-2, 2-6 = +30 -30 = 0. The DPV at the leaf of the tree specifies the DPV along the entire path from the root to that leaf.

Figure 2 shows a path from the root to a leaf of an attribute tree. Attributes a_1 to a_6 are on a path to leaf 32 for tree 30. The value of the product at leaf 32 is the sum of the attribute values $a_1, a_2, a_3 \dots a_6$, plus the base value.

IMPLICIT ATTRIBUTE TREES:

As mentioned above implicit attributes are those whose delta product values (DPVs) are not explicitly available in the data. Consider the example of an online airline reservation system like Travelocity. One can find prices for air tickets from City A to City B for different airlines, different arrival, departure dates and times, for different classes of services and for different number of stops between the two cities. An automatic information gathering system can repeatedly query the Travelocity server and obtain this information and store this information in a store as in a relational database. From the data stored in this store it can infer the attribute tree using complex sequel queries.

Figure 3 shows how the data from multiple data sources or applications can be obtained implicitly. A data gathering system 34 and integration system 38 populate relational store 36. Notice that any other kind of store may be used to store the information. In the example shown in figure 3, five pieces of data are stored for a particular airline in store 36 – from city, to city, time of day, day of week, fare. Using this information the Implicit Attribute Tree is constructed as follows. Heuristics may be used to build this tree efficiently. The table may look like this:

FROM	TO	TimeOfDay	DayOfWeek	Fare
SFO	NY	Morning	Monday	240\$
SFO	NY	Evening	Monday	200\$
ORL	SEA	Morning	Wednesday	110\$

The following steps may be followed in building the Attribute tree:

Step 1: The First values in the first two columns give the products. For instance, (SFO, NY) travel could be one product and (ORL, SEA) travel could be another product.

Step 2: Then the names of the columns (Attribute Metadata, in general) are identified and checked with the AM. This helps figure out the ranges and the intervals for that attribute. For instance, name of column 3 “Time of Day” (or the data type of the data in column 3) is checked with the Attribute Manager to determine the ranges and the intervals for that attribute.

Step 3: For each attribute whose range has m intervals a bit-vector of length m is created. For example, for DayOfWeek attribute a 7-bit vector is created with one bit for each day from Monday to Sunday. The I -th attribute value is represented by turning on the I -th bit in the bit vector.

Step 4: The bit vectors are concatenated in the sense that if there are N attributes each with range of length m a bit vector of length $N*m$ is created. This means that to represent and check all possible combinations of attribute values $m**N$ possible bit vectors may be created. Each row in the relational table shown above creates a new bit vector.

Figure 4 is a block diagram of an attribute-tree builder. Statistical system 48 reads relational data tables 40, 42, 44, which may be located on seller web sites. Online Analytical Processing (OLAP) system 46 may also be used to mine the data in relational tables 40, 42, 44 to determine delta values and attributes that determine the price of products in a product family. Systems 48, 46 then populate attribute trees 49.

To avoid creating an exponentially large tree, heuristics may be used to create a sparse, but yet, reasonable representative of the tree. Standard statistical sampling techniques

may be used to do this. Figure 5 shows one embodiment of a heuristic to create an attribute tree.

A table of fares (or values) is accessed, step 51. A parameter K is selected in step 52. K represents number of rows to select in steps 54, 55. A base value is selected, step 53. For instance, it may be the lowest fare in the table or just an average of all the fares in the table.

In step 54, K rows are selected from data store R that have the largest values. In step 55, the K rows with the lowest values are selected. Thus $2*K$ rows are selected from the table (store R) where K rows represent the top K fares and the other K represent the bottom K fares. K is fixed a priori in step 52.

The parameter P is selected, step 56. From the remaining rows, another P rows are randomly chosen based upon the values in columns corresponding to the leaf level nodes, step 57.

From the remaining rows, for each subtree rooted at the node above the leaves, representative rows are sampled based upon the variance of the data at the leaf level, step 58.

In step 59, the product values (DPVs) are computed for each path (each row picked from steps 54, 55, 57). The base product value must be carefully chosen in step 53. The tree is then re-balanced.

Note that other heuristics may be applied to produce a representative sparse enough tree of the big attribute tree. When such a tree is looked up for an attribute value and/or a Delta associated with it, either it is found or it is missed. If the tree is a sparse representative of the actual tree, the missing element may be just not represented in the tree. The lookup algorithm should return a Delta that can be “guess”ed from the sibling

of the node where the looked up attribute value could have been. In order to be able to do this, one needs to optimize the implicit attribute tree.

OPTIMIZING IMPLICIT ATTRIBUTE TREES – FIG. 6

Figure 6 is a procedure for optimizing an attribute tree. Optimizing the attribute tree keeps high variance nodes on top and low variance nodes at the bottom. It is preferable for the sake of reduced representation of the attribute tree that the attributes whose values have low variance appear at the bottom of the tree and the attributes whose values show high variance be at the top of the tree.

In order to do this the attributes are broken down into two types – Category types and Measure types. Measure types are those in which the values can be placed in a monotonic order. For instance, time-of-day, ages-of-people fall in this category. Category types include types like city-of-sale or shipping-company. Category types have values that are discrete in nature and there is no obvious monotonic ordering between attribute values. The choice of these types is somewhat arbitrary because while one can think of “city-of-sale” as a category type, one can look at the geographical location of the cities and identify a monotonic ordering based upon longitude or latitude. Category types can have hierarchies like city->state->country->continent or day->month->quarter->year.

One could run column-wise queries to do the following:

1. to identify the attributes that may be supported.
2. To identify the distribution of the data over that attribute type (steps 62, 64).

Once the distribution is identified, variance of the distribution can be identified, step 66. Then attributes with low variance can be moved towards the bottom of the tree and the attributes with higher variances can be moved towards the root of the tree, step 68. This is done by comparing columns pair-wise to see which columns results in an attribute that is the parent of the attribute created by the other column. Given two

columns C1, and C2, let A1 and A2 be the corresponding attribute types to be created. The node of type A1 is a parent of the one of type A2 if the following is true: The average of the variances of the C2 values for each C1 value is less than the average variance of the C1 values for each C2 value. (low variance nodes lower).

5

In the case where there are too many values for a category type data with hierarchy, it may be hoisted to a higher level datatype to reduce the number of attribute values. For instance, if there are 1000 different occurrences of city-of-sale over 10 states one may choose to keep only state-of-sale information. For Measure type data one may combine close measures into measure ranges. For instance, instead of keeping age information by age a programmer may decide to keep this information by age groups. This allows building compact representations of the attribute trees.

10

EXPLICIT ATTRIBUTE TREES:

As mentioned before Explicit Attribute trees are created based upon explicitly provided preferences of the TP. For instance, an air traveler might have the following decision logic:

15

If flight on Monday – delta = +30 else if on Sunday delta = -30 else if on Tuesday delta = +50

20

If flight in Morning delta = +30 else if in the afternoon delta = +50 else if at night delta = -10

...

25

From the logic given above an explicit tree may be created. It is preferable that the tree be sparse in nature for purposes of efficient processing.

CUSTOMIZING THE TRADING PARTNER TREE

The Trading Partner Tree can be transformed into simpler representations when a TP participates in a trade. The Trade Tree is a restriction of the TP tree where the TP is participating in a trade of a particular product where some of the attributes and attribute values are not required or supported.

USAGE OF THE ATTRIBUTE VALUE SYSTEM FOR INTELLIGENT E-COMMERCE APPLICATIONS

The TP Tree, implicit or explicit, is used by a Trading Partner (end user or business) in participating in an electronic commerce application. For instance, the inventors' system allows an electronic business to attach true values to items based upon the attributes specified by the user. It allows the businesses to employ buyer-specific pricing of items based upon the attributes specified by the buyer. It also allow businesses to target-sell items based upon user preferences.

Figure 7 is a flowchart for a Business-to-Consumer (B2C) application of the attribute trees. User visits a Vendor site to buy a product P, step 70. The user provides specification for product P along with a list of attributes and values that he/she cares about, step 72.

The user specifies a baseline amount that he is willing to pay for the product, steps 74, 76. (This is an optional block).

The user may optionally specify how much he values each attributes, step 78. In one embodiment the user specifies "Delta" amounts for zero or more of the attribute and attribute values, steps 80, 82, 84.

The vendor identifies and sells the product instance that the prospective buyer values the most (the one that computes to the maximum on the sum of the "Delta" values on

all the attributes; an alternative to the “Delta” amounts may be used like some other numeric scheme negotiated between the buyer and seller). This is shown in steps 86, 88, 89.

5 The vendor could even dynamically place a true value on the product instance based upon the buyer attribute preferences.

AUCTION SCENARIOS

The attribute system enhances electronic auction systems where the bidder and the seller can price and/or determine the value of an auction item based upon multiple attributes of the item wherein the overall value of an item is determined by the value attached to individual attributes related to the item. In a forward auction system there is a single seller and multiple prospective buyers.

Figure 8 is a flowchart of an auction system enhanced by the attribute-structuring facility. Given a list of attributes associated with a product being auctioned, the method of figure 8 helps a prospective buyer to choose a bid based upon the attributes that he/she values the most. The seller puts up an item for auction with its attributes and attribute values, step 90. Each bidder has a baseline value for the item, and the maximum bid prices is set to the baseline value, steps 92, 94. For each attribute that the bidder cares about, step 96, the bidder has a delta to be added/subtracted if the product includes/excludes that attribute, step 98. By going through all the attributes of the product (steps 95, 96) and adding/subtracting the deltas from the baseline value, the system computes the maximum bid price for the bidder, step 99. The bidder bids on the product between the baseline and the maximum bid price for what he/she thinks is the real value of the product.

Figure 9 is a flowchart of an optimal product/seller selector that helps a bidder. In an auction system, where multiple products are being auctioned and each of the products have listed associated attributes, it enables the prospective buyer to bid on items that

satisfy most of his/her attributes. Here the bidder can pick between multiple sellers who are selling “similar” products but different support for attributes that the bidder cares about. The bidder looks for a product whose attributes have the most overlap with the attributes the bidder cares about and bids on that/those product(s).

5

A bidder B_i looking for a product P finds several sellers auctioning the product in various configurations, steps 102, 104. For each seller, the seller’s attribute tree is compared to the buyer’s attribute tree, steps 108, 109. The attribute names, values, and ranges may not exactly match, so they may need to be normalized. The degree of overlap detected in step 110 is stored for that seller. The attribute tree comparison is repeated for the other sellers with the same buyer’s attribute tree, steps 112, 106-110. The maximum overlap is found, step 114. The maximum overlap indicates an optimal product configuration by one of the sellers that most closely matches the desired configurations of the buyer. The buyer can then bid or purchase the product with the optimal configuration from the most-closely overlapping seller, step 116.

10

15

A selling party may provide augmented products where the product being auctioned is dynamically augmented with new attributes to satisfy the attributes that are valued the most (or by most prospective buyers). Here the seller goes through all the attributes that bidders for a particular product care about and augments the product with those attributes and values that make it the most marketable item.

20

Figure 10 illustrates dynamic augmentation of a product with the attributes most desired by the potential buyers. A matrix of attributes and possible attribute values is created for a product, step 130. Preference data is collected from several prospective buyers or bidders, step 132. For each potential buyer B_i , each attribute is checked to see if the user has entered a preference or attribute value for that attribute, step 138. When the buyer has entered an attribute value V_{jt} for attribute A_j , a counter for that attribute value is incremented, step 140.

25

30

An inner loop increments the loop variable j for all attributes, steps 142, 144. An outer loop increments the loop variable i , for all potential buyers, steps 146, 148. The loop variables are initialized in steps 134, 136. Thus all variables are checked for all potential buyers. The attribute counters incremented in step 140 are compared to find the maximum count(s), and the corresponding attributes are used to augment the product, step 150. The maximum counts corresponds to the most-desired attributes for the product.

The above scenarios are only representative uses of the inventors' system in some auction systems. Many other scenarios and uses are contemplated.

AGGREGATION

Aggregation is used in auctions to aggregate small buyers and sellers to match them up with a seller or a buyer who can only sell to or buy from large parties. Attributes can enhance the aggregation process by allowing aggregation of buyers and sellers not just based on products but based upon similarity in the attributes they value the most.

Java interface classes and pseudo code were attached to the provisional application.

ADVANTAGES OF THE INVENTION

The inventors' multi-attribute model is well-suited for electronic exchanges. The attribute trees provide a way to express the many options or configurations of products in a product family. The attribute tree is a convenient mechanism to receive, store, and compare multiple product configurations from many buyers and sellers. The attribute tree is a simple yet powerful structure for storing many attributes of a product family.

The invention provides a system that allows a trading partner to model a product or augment a product with multiple attributes statically or dynamically based upon the other participants in the trade. The system discovers attributes from implicit set of data,

optimizes attribute representation when large number of attributes are present, and also allows dependencies between attribute values. The system can be used in a Business-to-consumer or Business-to-Business scenario.

ALTERNATE EMBODIMENTS

Several other embodiments are contemplated by the inventors. The increment of the counter can be by more than 1, or a negative increment (decrement) can be substituted. Non-binary counting schemes such as Gray Code can be substituted. Rather than incrementing the counter, the value that the buyer places on an attribute can be summed. Thus buyers who place more value on an attribute than other buyers can have a larger weighting for that attribute when the optimal set of attributes is determined.

Attributes may also be discovered from the preferences of the trading partners. For example, a traveler may specify that he “prefers” flying in the morning and “detests” flying at night. These may be used to express time-of-day attributes with the value of “morning” with a positive delta of 20% of the base value of a ticket and the value of “night” with a negative delta of 50% of the base value. The attribute value of “afternoon” may have a delta of 0.

The foregoing description of the embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

We Claim:

1. A multi-attribute valuing system comprising:

5 a buyer attribute-input for receiving a plurality of delta values for a product, each delta value indicating a higher or a lower true-value that the buyer places on the product when modified by the attribute associated with the delta value;

a seller attribute-input for receiving a plurality of delta values for the product, each delta value indicating a higher or a lower true-value that the seller places on the product when modified by the attribute associated with the delta value;

10 an attribute manager, receiving the attributes and delta values from the buyer and from the seller, for creating and managing attribute trees, each attribute tree specifying dependencies among the attributes of the product, the attributes associated with delta values that adjust a true-value of the product;

15 an attribute-tree analyzer, receiving an attribute tree for the buyer and an attribute tree from the seller for the product, for comparing the delta values from the buyer with the delta values from the seller for a plurality of attributes that modify the product, the attribute-tree analyzer selecting an optimal specification of the product, the optimal specification including a series of the attributes; and

20 a product specifier, coupled to the attribute-tree analyzer, for reporting to the seller the series of attributes for the optimal specification of the product,

whereby the optimal specification of the product is determined by analyzing delta values specified by the buyer and the seller that adjust the true-value of the product.

25 2. The multi-attribute valuing system of claim 1 wherein each branch of the attribute tree specifies an attribute, including an attribute name, an attribute value, and a delta value,

whereby the attribute value specifies a value of the attribute that modifies the product itself, while the delta value specifies a true-value change offered to trade the product.

3. The multi-attribute valuing system of claim 2 wherein each branch of the attribute tree also specifies one or more ranges where each range has one or more interval of attribute values.

5 4. The multi-attribute valuing system of claim 2 wherein each attribute tree includes a root, the root for specifying a base product unmodified by the attributes;

wherein a path from the root to a leaf of the attribute tree specifies a final product,
 wherein the final product is the base product modified by attributes along the
 10 branches in the path from the root to the leaf,
 whereby the path through the attribute tree specifies attributes for the final product.

5 5. The multi-attribute valuing system of claim 4 wherein the root of the attribute tree includes a baseline value, the baseline value being a value for the base product;

wherein each node in each branch of the attribute tree includes the delta value, the delta value being a value adjustment for the product modified by the attribute specified by the branch;

wherein a final true-value of the final product is generated by adding the baseline value to a sum of all delta values along a path in the attribute tree from the root to a
 20 leaf for the final product,

whereby the final true-value is determined by summing the baseline value with the delta values along a path in the attribute tree.

25 6. The multi-attribute valuing system of claim 2 wherein the product is an item for sale or a service.

7. The multi-attribute valuing system of claim 2 wherein the attributes and the delta values are explicitly defined by the buyer or seller.

8. The multi-attribute valuing system of claim 2 wherein the attributes and delta value are implicit, the delta values being determined from other data.

9. The multi-attribute valuing system of claim 8 further comprising:

5 a data gatherer, coupled to a seller web site, for reading other data from the seller web site, the other data including price data for products with various attributes;
a data store, coupled to the data gatherer, for storing the other data;
an implicit tree builder, coupled to read the other data in the data store, for generating
the attribute tree for the seller web site by comparing the price data in the other
10 data for products with different attributes,
whereby the attribute tree is generated implicitly.

10. The multi-attribute valuing system of claim 9 wherein the data store includes records that contain a product specifier, attributes for the product, and a value for the product with the attributes stored in the record.

11. The multi-attribute valuing system of claim 10 wherein the data store is a relational database.

12. The multi-attribute valuing system of claim 9 further comprising:
heuristic means for re-balancing the attribute tree generated by the implicit tree builder
to produce a sparse representation of the attribute tree,
whereby the attribute tree is re-balanced.

13. The multi-attribute valuing system of claim 12 further comprising:
optimizing means, coupled to the attribute tree, for re-arranging attributes in the
attribute tree, wherein attributes with greater variances in attribute values are
moved toward the root while attributes with lower variances are moved toward
the leaf,

whereby high-variance attributes are moved near the root of the attribute tree during optimization.

14. A computer-program product comprising:

5 a computer-usable medium having computer-readable program code means embodied therein for indicating differential values of products in a product family, the computer-readable program code means in the computer-program product comprising:

attribute means for storing product-attributes that specify products within the product family, the products varying according to product-attributes;

10 input means for receiving delta values from a user, the delta values representing differences in values of products in the product family, the products varying according to the product-attributes, the delta values representing differences in value perceived by the user for the product specified by the product-attributes; and

value attach means, coupled to the input means and to the attribute means, for attaching the delta values to the product-attributes to form a user attribute-store,

whereby the user attribute-store indicates differences in values perceived by the user for products in the product family that vary in specification according to the product-attributes.

15. The computer-program product of claim 14 wherein the product-attributes include dependencies wherein a value of one product-attribute depends on a value of a second product-attribute, the attribute means including dependency means for storing the dependencies of the product-attributes for the product family,

whereby product-attribute dependencies are stored.

16. The computer-program product of claim 15 wherein the user attribute-store comprises an attribute tree having a tree-like logical structure, the attribute tree

having a root that specifies a base product in the product family, the root branching into nodes according to the product-attributes, the final terminal nodes being leaves, the leaves each representing a product in the product family, whereby the user attribute store is an attribute tree.

5

17. The computer-program product of claim 16 wherein a differential value of a product is determined by summing delta values for each node along a path from the root to the leaf for the product, the final value of the product being the differential value added to a base value for the base product,

10 whereby product values are determined by summing delta values along a path in the attribute tree.

18. The computer-program product of claim 14 wherein users include buyers and sellers of a product in a product family;

15 wherein a buyer is associated with a buyer attribute-store that is a user attribute-store that stores the delta values perceived by the buyer;

wherein a seller is associated with a seller attribute-store that is a user attribute-store that stores the delta values perceived by the seller;

whereby buyer and seller delta values are stored by user attribute-stores.

19. The computer-program product of claim 18 wherein the users also include facilitators and market makers, wherein trading partners include buyers, sellers, facilitators and market makers.

25 20. The computer-program product of claim 18 wherein the computer-readable program code means further comprises:

a value comparator, receiving the buyer attribute-store and the seller attribute-store, for comparing the delta values from the buyer attribute-store and the seller attribute-store for each product-attribute to find an optimal group of product-attributes

30 that specify an optimal product for sale by the seller to the buyer,

whereby an optimal product in the product family for a sale is determined by comparing attribute stores of delta values for the buyer and for the seller.

21. The computer-program product of claim 20 wherein the buyer and seller are
5 connected by a network;

wherein the computer-readable program code means further comprises:

network connection means for transmitting and receiving packets from remote clients
for the buyer and the seller,

whereby an electronic exchange connects the buyer and the seller.

22. The computer-program product of claim 14 wherein the input means includes an
explicit input to receive delta values from the user,
whereby the delta values are directly input by the user.

23. The computer-program product of claim 14 wherein the input means includes an
implicit input that reads full-values for products in the family, the input means
including compare means for determining the delta values for the product-
attributes that specify the products in the product family,
whereby the delta values are implicitly input from the user.

24. The computer-program product of claim 23 wherein the input means includes
scan means for scanning a product and value list to implicitly input the delta values.

25. A computer-implemented method for selecting a product for sale to a trading
25 partner comprising:

receiving a selection of a base product from the trading partner;

receiving a list of attributes for the base product from the trading partner;

for each attribute in the list of attributes, receiving a delta value from the trading
partner for the attribute, the delta value indicating an additional value the

trading partner places on the product when the base product is modified by the attribute, the delta values being trading partner-specified;
 selecting as a selected product the product modified by a subset of the attributes in the list of attributes, the subset of the attributes including attributes wherein a sum
 5 of the delta values for attributes in the subset of the attributes is a selected sum;
 offering the selected product to the trading partner for sale to the trading partner;
 whereby the product sold is selected using a selected sum of the delta values of the attributes specified by the trading partner.

10 26. The computer-implemented method of claim 25 further comprising:
 receiving a baseline value from the trading partner, the baseline value indicating a value the trading partner places on the base product when not modified by any of the attributes;
 wherein the baseline value from the trading partner is an amount the trading partner is
 15 willing to pay for the base product, and the delta values are additional amounts the trading partner is willing to pay for the product modified by the attributes,
 whereby attributes of the selected product for sale to the trading partner are selected based on value specified by the trading partner for the attributes.

20 27. The computer-implemented method of claim 26 further comprising:
 dynamically generating a true-value for the selected product based on the selected sum of the delta values and the baseline value,
 whereby a sales price is adjusted for trading partner-specified delta values of attributes for the selected product.

25 28. The computer-implemented method of claim 25 further comprising:
 creating a trading partner attribute tree from the list of attributes and the delta values received from the trading partner, the attribute tree specifying values to the trading partner of the product modified by the attributes,

whereby the trading partner specifies values of the product modified by different attributes.

29. The computer-implemented method of claim 28 further comprising:

5 making a connection to the trading partner over the Internet.

30. A computer-implemented tool for finding an optimally-specified product for purchase comprising:

10 a buyer attribute-tree generator for generating a buyer attribute tree, the buyer attribute tree having a base product at a root of the buyer attribute tree, wherein each branch in the buyer attribute tree specifies an attribute that modifies the base product;

base-product means for searching for sellers that are offering the base product for sale, the base-product means generating a sellers list of sellers that offer the base product for sale;

15 a seller attribute-tree generator for generating a plurality of seller attribute trees, each of the seller attribute trees having the base product at a root of the seller attribute tree, wherein each branch in the seller attribute tree specifies an attribute that modifies the base product;

20 an attribute tree comparator, receiving the buyer attribute tree and one of the seller attribute trees, for comparing the buyer attribute tree to the seller attribute tree and determining an overlap, the overlap indicating a degree of overlap of the attributes for the base product;

25 wherein the attribute tree comparator compares seller attribute trees to the buyer attribute tree for all sellers in the sellers list; and

an overlap maximizer, coupled to the attribute tree comparator, for finding a maximum overlap generated by the attribute tree comparator, the maximum overlap being generated for a maximum seller attribute tree of a chosen seller;

30 wherein the chosen seller offers for sale the base product with attributes most-closely matching attributes specified by the buyer in the buyer attribute tree,

whereby the seller is chosen based on maximum attribute overlap by comparing attribute trees.

31. The computer-implemented tool of claim 30 further comprising:

a linker, responsive to the chosen seller from the overlap maximizer, for connecting the buyer with the chosen seller to allow the buyer to purchase the base product with the attributes most-closely matching buyer-specified attributes.

32. The computer-implemented tool of claim 31 further comprising:

a bid generator, coupled to the chosen seller, for sending a bid from the buyer to the chosen seller, the bid including a specification of the base product and the attributes that overlap the buyer attribute tree and the seller attribute tree for the chosen seller.

33. A method for dynamically augmenting a product with attributes that are most valued by prospective buyers comprising:

receiving a plurality of attributes and attribute values for the attributes from the prospective buyers;

for each attribute, and for each prospective buyer,

determining when a buyer has submitted an attribute value for the attribute;
when a buyer has submitted the attribute value, adjusting a counter for the attribute value;

examining the counter for each attribute value to find maximum attribute values, the maximum attribute values having the most prospective buyers submitting the attribute value to indicate buyer preference of the attribute with the attribute value;

modifying the product according to the attributes with the maximum attribute values to form an augmented product that is more valued by the prospective buyers; and offering the augmented product for sale to the prospective buyers,

whereby the product is augmented with attributes preferred by most of the prospective buyers.

34. The method of claim 33 wherein adjusting the counter for the attribute value
 5 comprises incrementing the counter for each prospective buyer that submitted the attribute value,

whereby the counter is incremented for each prospective buyer submitting the attribute value for the attribute.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208

ATTRIBUTE TREE FOR MODELING VALUE OF MULTI-ATTRIBUTE
PRODUCTS/SERVICES FOR AN ONLINE TRADING SYSTEM

5

ABSTRACT OF THE DISCLOSURE

10

An electronic commerce system facilitates trading of product families that contain individual products with varying attributes. The product's attributes specify products within the product family that vary according to these attributes. Delta values are received from a user. The delta values represent differences in values of products in the product family. These differences in value are perceived by the user for each of the products specified by the attributes. The delta values are attached to the attributes to form a user attribute tree. The root of the attribute tree is the base product, while the leaves of the attribute tree represent various configurations of the product. Summing delta values along a path from the root to a leaf provides the additional value of that product configuration. A buyer's attribute tree contains the buyer's perceived values for many different product configurations. The buyer's attribute tree can be compared to a seller's attribute tree. The overlap of the buyer's and seller's attribute trees can be used to find an optimal product configuration for sale to the buyer.

5

20

Airline Trading Partner Tree

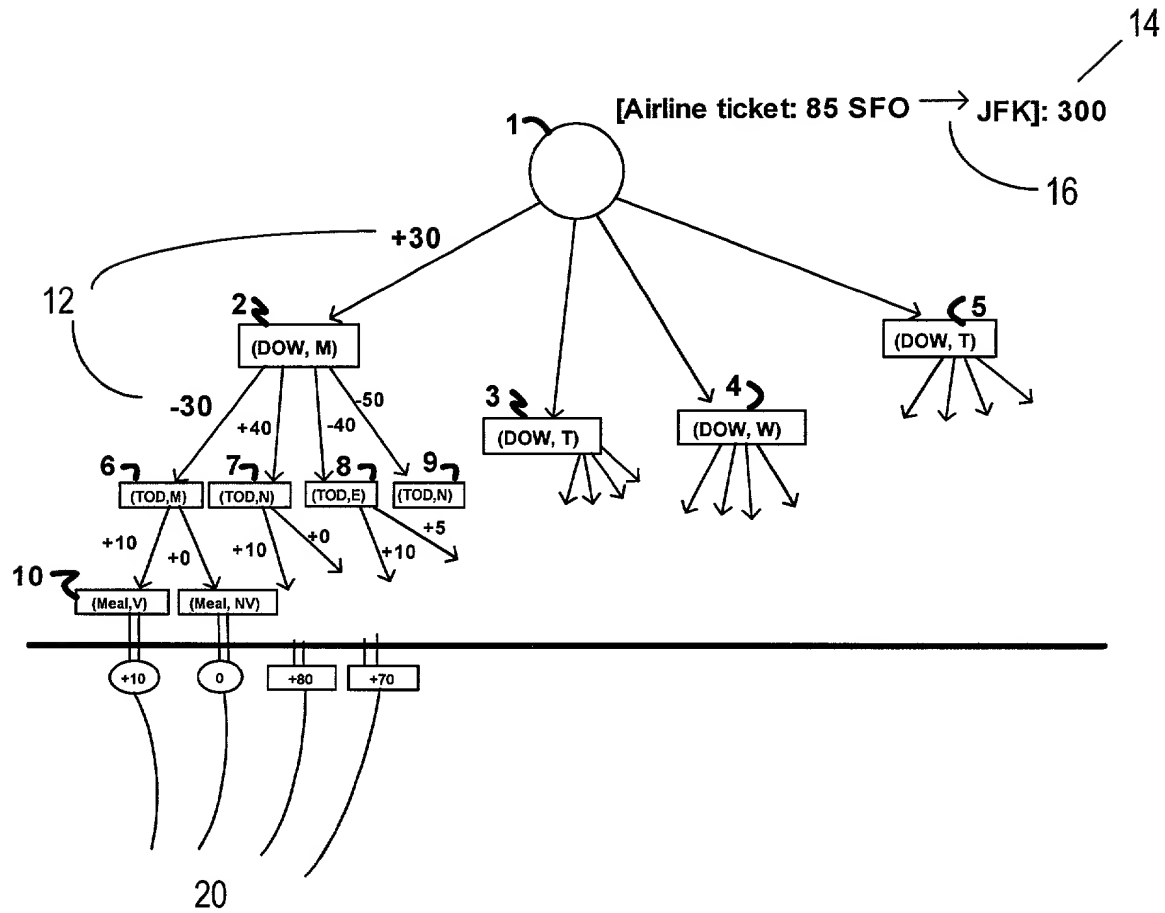
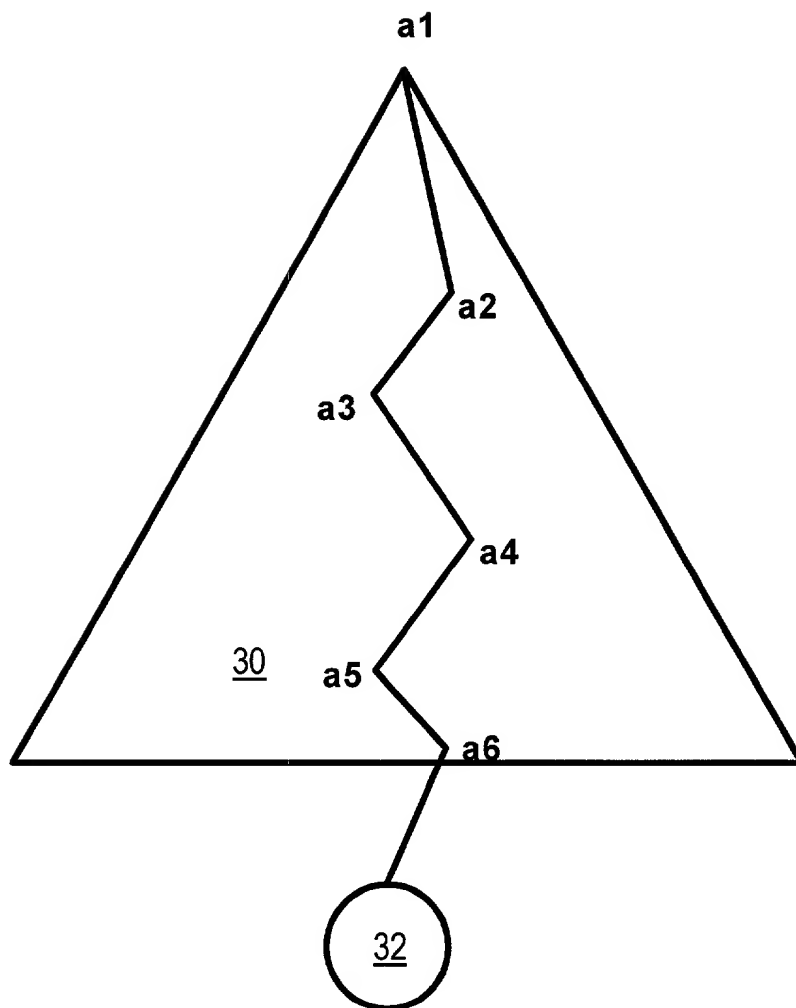


Figure 1



S - Product Value
PRODUCT $\langle a_1, a_2, a_3, \dots, a_6 \rangle$

Figure 2

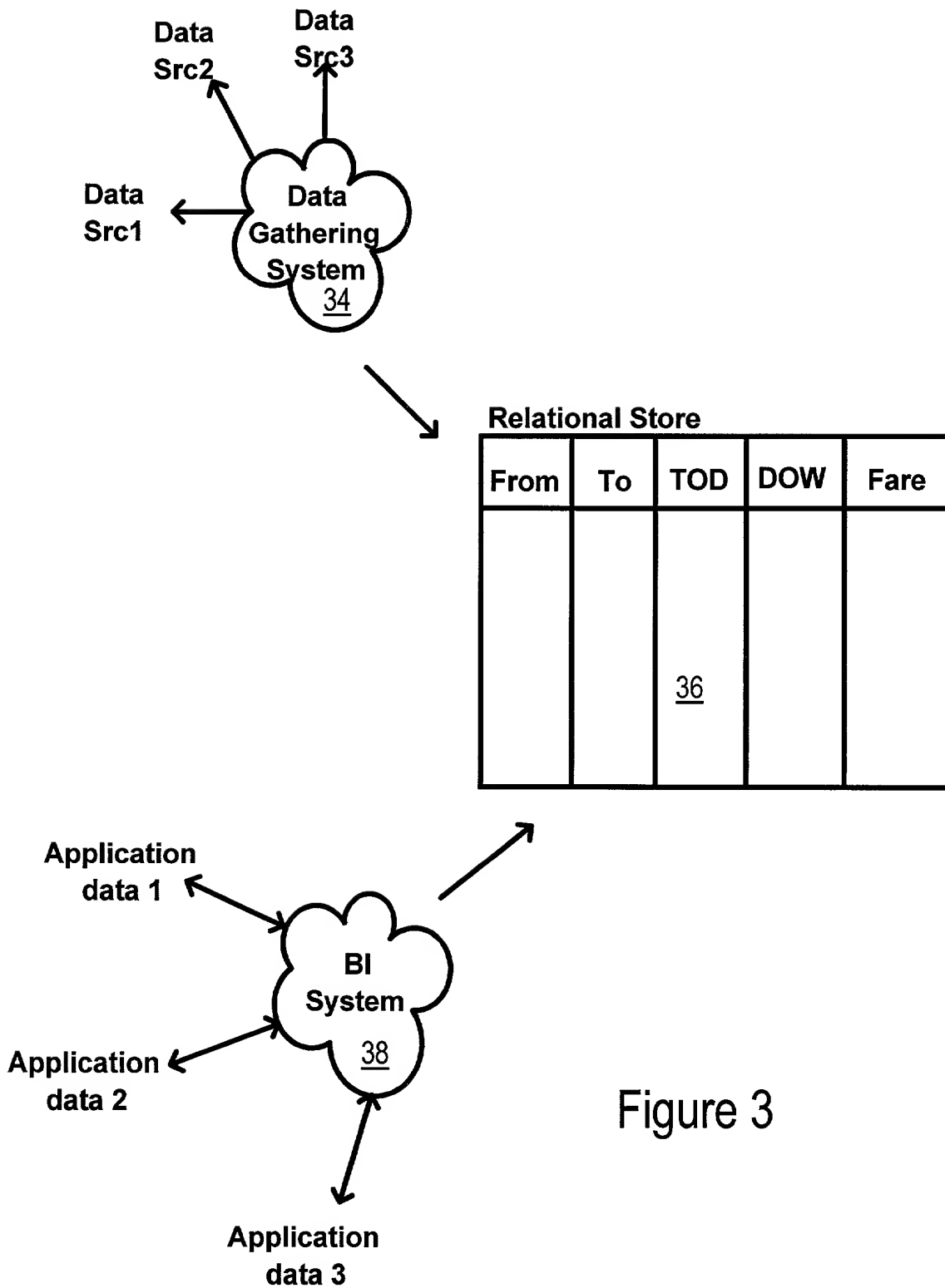


Figure 3

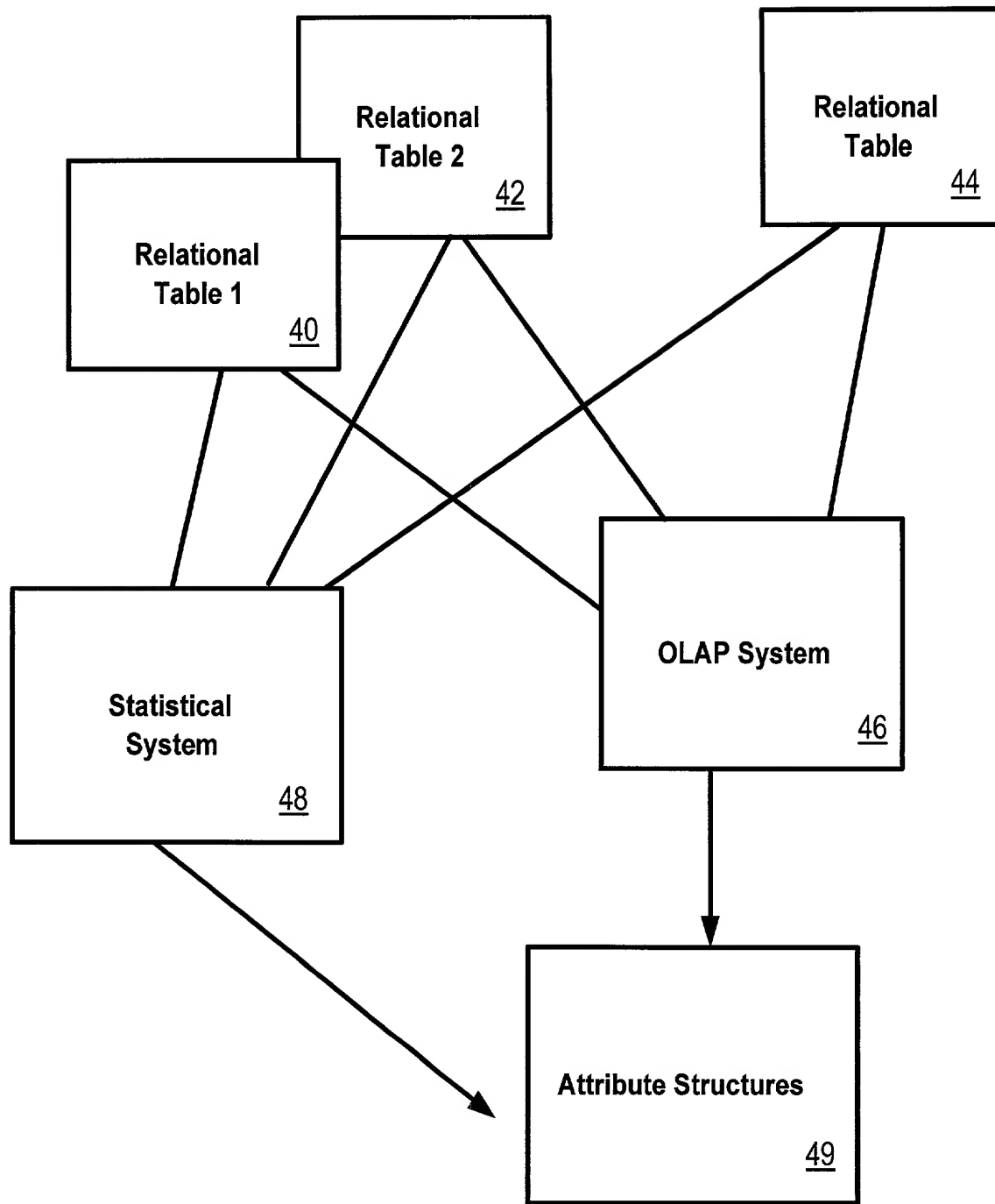


Figure 4

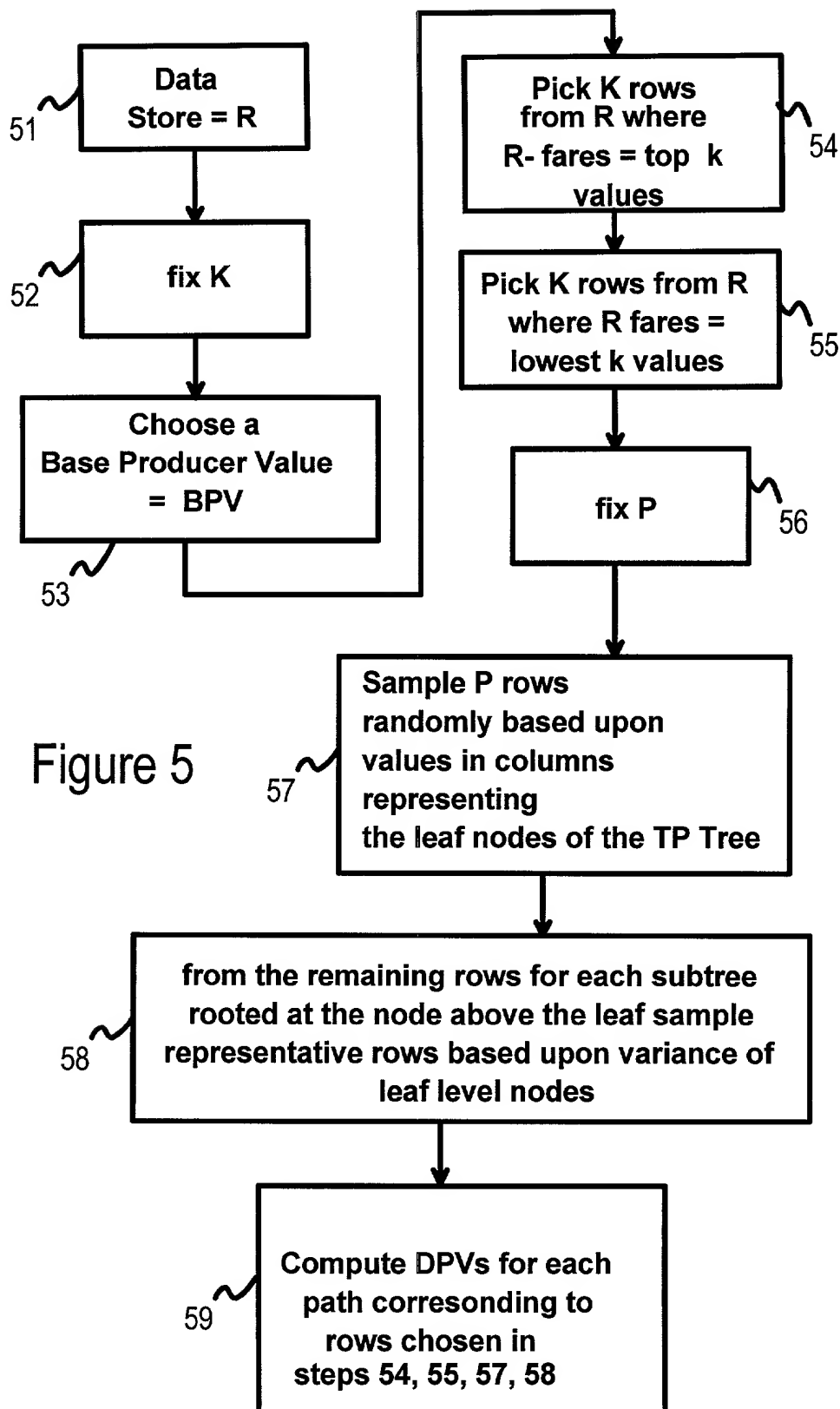


Figure 5

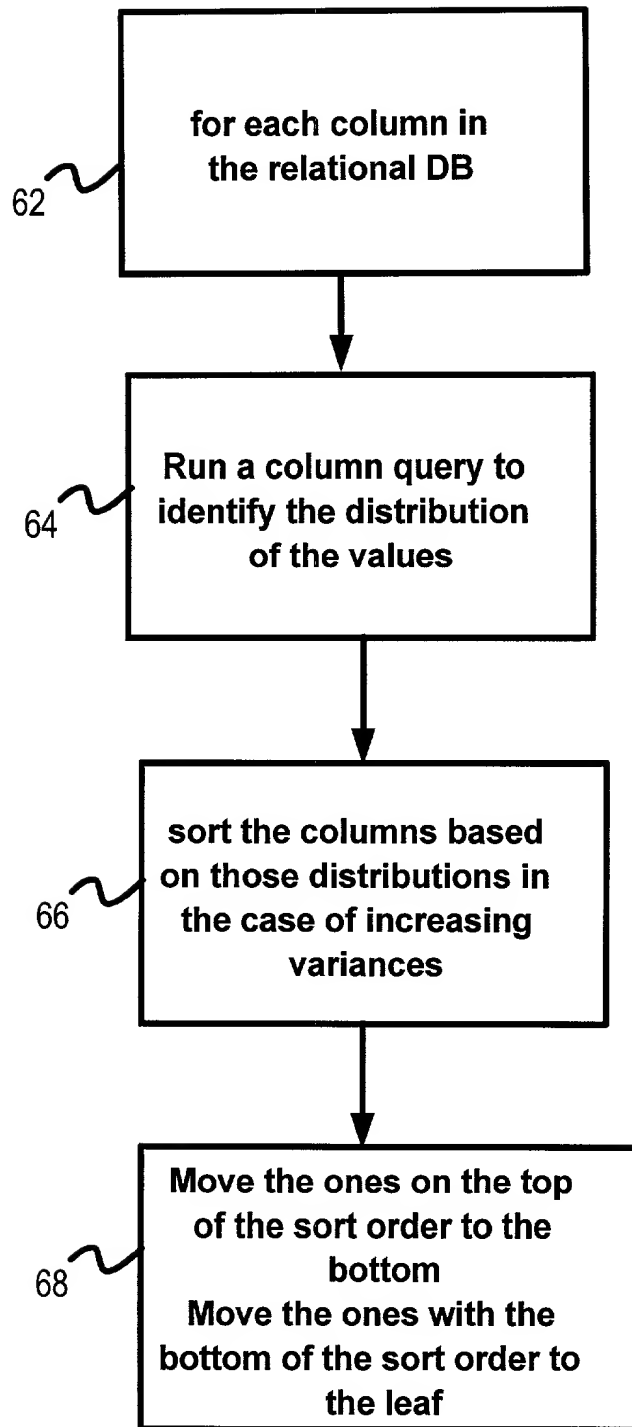


Figure 6

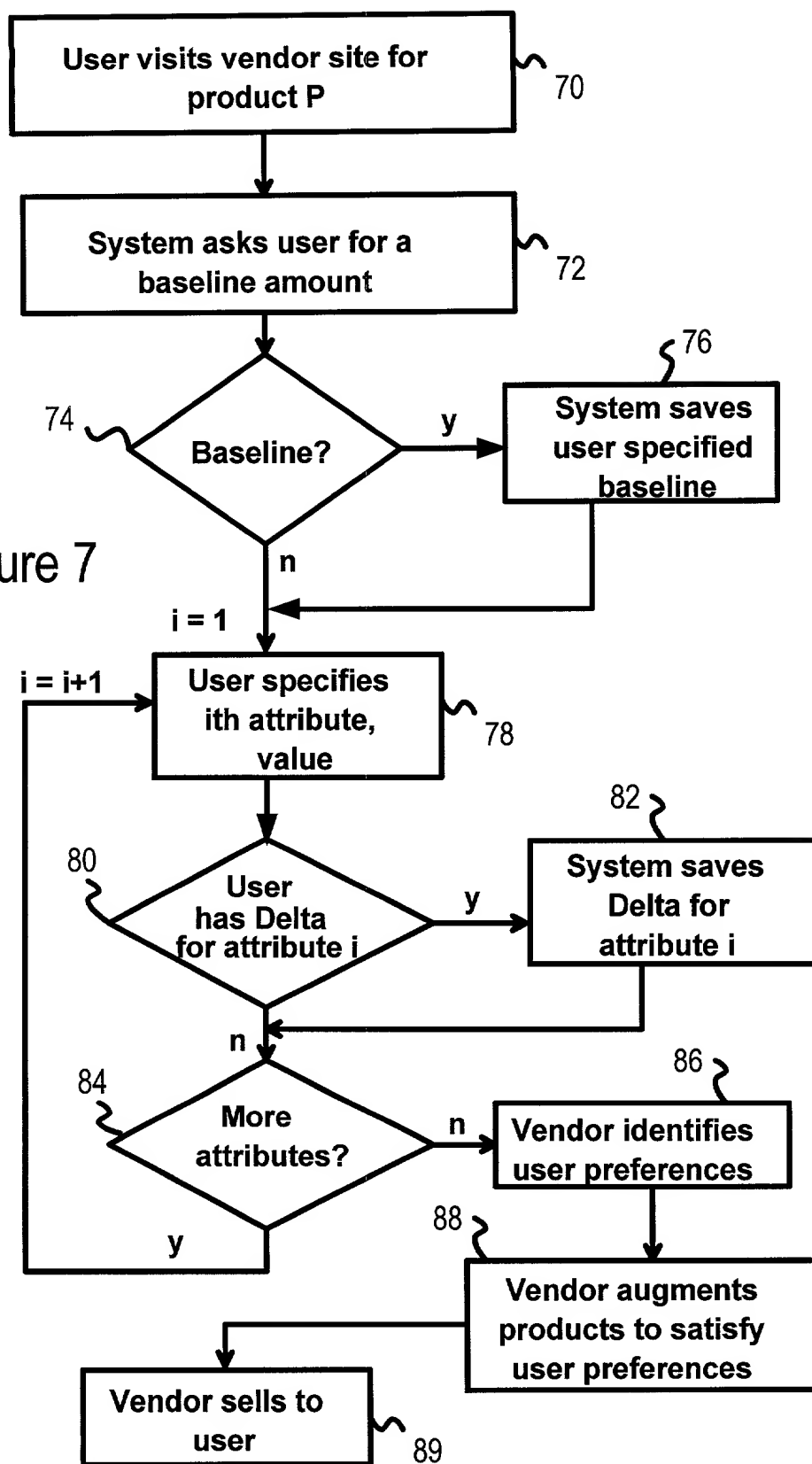
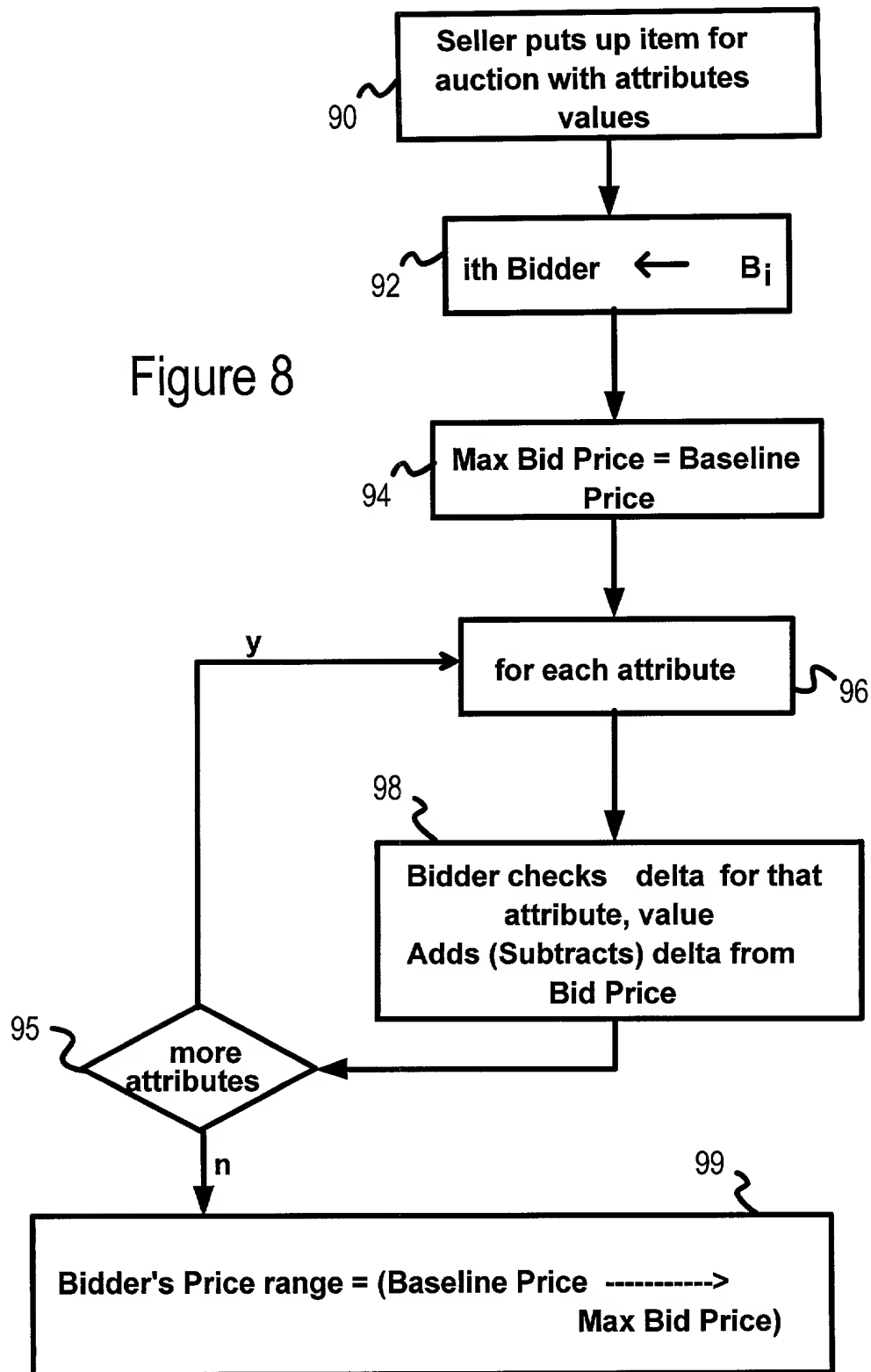
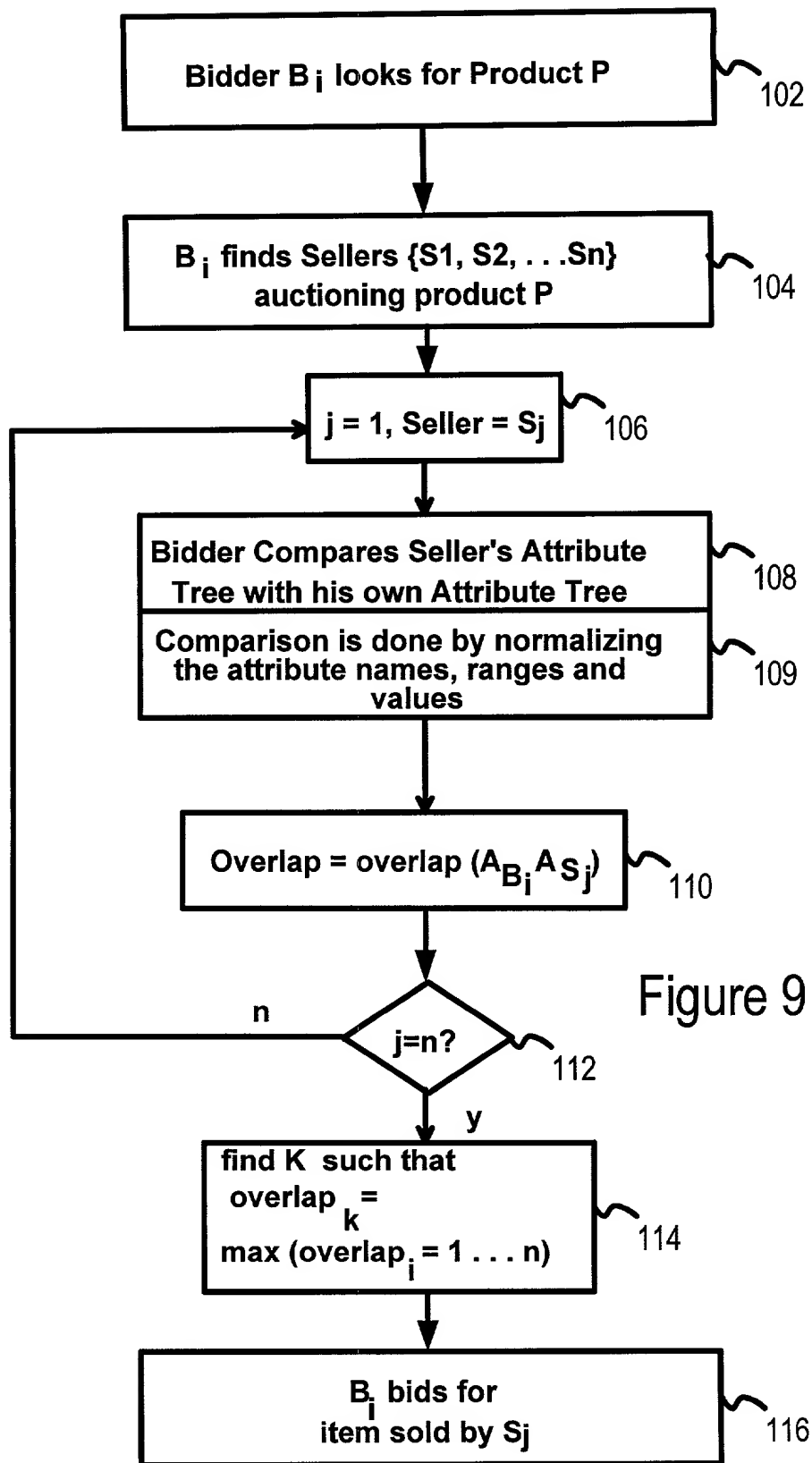


Figure 8





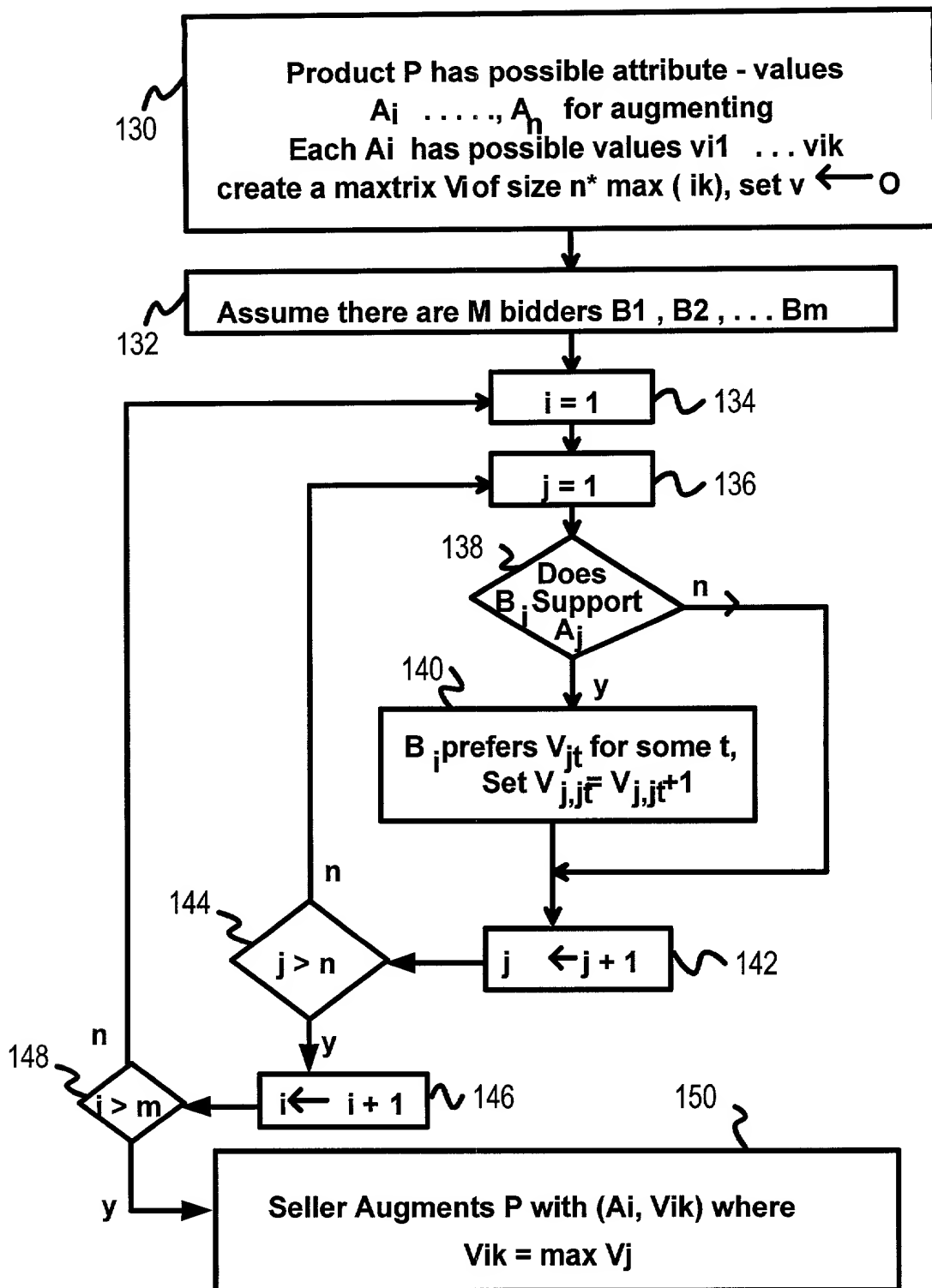


Figure 10

DECLARATION FOR PATENT APPLICATION

NH-1

As a below-named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe that I am an original, first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention, the specification of which is attached hereto and which has the following title:

Attribute Tree for Modeling Value of Multi-Attribute Products/Services for an Online Trading System

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment specifically referred to in the oath of declaration. I acknowledge a duty to disclose information which is material to examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby appoint the following Agent to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Stuart T. Auvinen, Reg. No. 36,435

Address all telephone calls to Stuart T. Auvinen at telephone No. (831) 476-5506.

Address all correspondence to:

Stuart T. Auvinen
429 Twenty-Sixth Ave.
Santa Cruz, CA 95062

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, United States Code, Section 1001, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Joint/First Inventor's Signature Ravi V. Condamoor date: May 24, 00

Full name of first inventor: Ravi V. Condamoor
Residence: Sunnyvale, CA Citizen of: USA
Post Office Address: 1382 FisherHawk Drive, Sunnyvale, CA 94087

Joint/Second Inventor's Signature Ankur Datta Sharma date: 5/24/2000

Full name of second inventor: Ankur Datta Sharma
Residence: Sunnyvale, CA Citizen of: USA
Post Office Address: 1065 Marigold Court, Sunnyvale, CA 94086

Joint/Third Inventor's Signature Neelakantan Sundaresan date: 05/24/2000

Full name of third inventor: Neelakantan Sundaresan

Residence: San Jose, CA

Citizen of: India

Post Office Address: 492 Capitol Village Circle, San Jose, CA 95136